Navigating a corpus of journal papers using Handles

Christopher Chapman & David F. Brailsford
Electronic Publishing Research Group
School of Computer Science and IT
University of Nottingham
NOTTINGHAM NG8 1BB
UK

ABSTRACT

For some years now the Internet and World Wide Web communities have envisaged moving to a 'next generation' of Web technologies by promoting a globally unique, and persistent, identifier for identifying and locating many forms of 'published objects'. These identifiers are called Universal Resource Names (URNs) and they hold out the prospect of being able to refer to an object by *what* it is (signified by its URN), rather than by *where* it is (the current URL technology). One early implementation of URN ideas is the Unicode-based Handle technology, developed at CNRI in Reston Virginia. The Digital Object Identifier (DOI) is a specific URN naming convention proposed just over 5 years ago and is now administered by the International DOI organisation, founded by a consortium of publishers and based in Washington DC. The DOI is being promoted for managing electronic content and for intellectual rights management of it, either using the published work itself, or, increasingly via *metadata* descriptors for the work in question.

This paper describes the use of the CNRI handle parser to navigate a corpus of papers for the *Electronic Publishing* journal. These papers are in PDF format and based on our server in Nottingham. For each paper in the corpus a metadata descriptor is prepared for every citation appearing in the *References* section. The important factor is that the underlying handle is resolved locally in the first instance. In some cases (e.g. cross-citations within the corpus itself and links to known resources elsewhere) the handle can be handed over to CNRI for further resolution.

This work shows the encouraging prospect of being able to use persistent URNs not only for intellectual property negotiations but also for search and discovery. In the test domain of this experiment every single resource, referred to within a given paper, can be resolved, at least to the level of metadata about the referred object. If the Web were to become more fully URN aware then a vast directed graph of linked resources could be accessed, via persistent names. Moreover, if these names delivered embedded metadata when resolved, the way would be open for a new generation of vastly more accurate and intelligent Web search engines.

1 INTRODUCTION

One of the most striking recent developments in electronic data interchange has been the specification of XML [1, 2] as an 'enabling metasyntax' and its subsequent enthusiastic adoption on the World Wide Web and elsewhere. XML is rapidly transforming every aspect of the way digital documents are internally structured and how they are externally accessed via hyperlinks.

Metadata (i.e. 'data about data') for 'publications'—in the widest sense—has been collected, somewhat haphazardly, by commercial and academic publishers for many years. A secondary publishing industry has developed via companies such as ISI, Science Citation Indexes and BioMedNet based on creating and selling metadata for scientific papers (e.g. metadata items such as Title, Author, Date, Abstract, Keywords etc.). Given that metadata is usually well structured, human-readable and plain text in nature, it is natural to want to use XML to represent it.

The World Wide Web Consortium (W3C) answered calls for an XML metadata framework by creating the Resource Description Framework, usually known by its abbreviation of RDF. Using RDF, complex and structured metadata can be specified in XML syntax. RDF is capable of representing a directed graph of properties and relationships but the underlying XML syntax ensures that the basic ability to parse RDF is already present in the latest generation of XML-based software.

The next section examines how the linking of a URN to metadata about a digital object – as a staging point before a possible linking to the object itself – can confer great benefits in searching for, and acquiring, information on the Web.

2 PERSISTENT IDENTIFIERS AND METADATA

An area of Internet and Web development that has received much attention is that of URNs (Universal Resource Names) or 'persistent identifiers'. Like the familiar URL, a URN is a member of a wider set known as Universal Resource Identifiers (URIs) [3]. But while URLs refer to a file's physical location, URNs are intended to be independent of such details.

The URN paradigm is to call objects by a unique name. The name chosen need not be inherently meaningful, but there are obvious advantages if this is the case. The additional requirement is that URN identifiers shall be global and persistent – they will, essentially, never expire. URNs can also utilise existing naming schemes, such as ISBN, simplifying the transition of legacy systems to the new technology.

The subject of URNs is still surrounded by considerable, and often heated, debate. Much of this discussion centres on the scalability of URN implementations, given that the extra burden on the Domain Name Service (DNS), and ultimately on hardware such as routers, will be considerable if a future DNS has to query servers with a URN to find a 'nearby' instance of an object rather than following a step-by-step resolution of a URL to a specific server. Controversy rages also on what the permitted subset of characters should be that can be used by a URN. The handle implementation of a URN, used in this paper, allows the full Unicode character set to be used but it has been argued [4] that exotic characters chosen from tens of thousands of possibilities are not a good foundation for maintainable and persistent identifiers; the restricted 7-bit ASCII subset would, perhaps, be a much more stable basis. Ultimately these discussions will produce results that have far-reaching implications for future URN development but for the purposes of our investigations we wanted a persistent identifier system that was sufficiently well developed to use in a real application, but with sufficient flexibility that the principles of our research could be easily transferred to any future URN system.

The Handle system, developed at the Corporation for National Research Initiatives (CNRI) in Reston, Virginia, is one such system [5]. It is described as a global, distributed name resolution system. A handle is a persistent, global identifier, and looks something like:

```
hdl:1034/example-handle
```

Note that `hdl:` is the name of the protocol, not part of the handle itself. It is analogous to the `http:` part of a URL. The `1034` is the handle code that has been assigned to Nottingham, and is unique to our institution.

The remainder of the identifier is the name of the particular handle in our namespace. Since we are a handle naming authority (number 1034), we have full control over this namespace, and the handles we create within it. This means we can create our own naming scheme, or migrate from an existing scheme, for our part of 'handle space'.

In some ways, similarly to the DNS, the Handle system resolves the name of a handle (the address) to data. Unlike DNS, however, which resolves to a single IP address, a handle resolves to an arbitrary series of name/value pairs. These data can be considered the *contents* of the handle itself.

Any type of data can be stored within a handle, including metadata. The advantages are clear – using DNS, a single identifier (a URL) would be resolved to a single file somewhere on a server. Using handles, we can resolve a handle identifier to one or more files (or even multiple copies of the same file on different servers), to administrative information, and also to document metadata. Because the metadata is stored *as a part of the handle itself*, it can be accessed separately from the document. Indeed, a handle can be created for a document that does not exist in electronic form. The handle for such a document would probably not contain any URLs, since there is no electronic document to redirect to, but the metadata could still be present – and this has value in itself.

Thanks to the software and libraries developed at CNRI, the Handle system is sufficiently mature to use in applications. At Nottingham we run a handle server, written in Java by CNRI and freely downloadable from their website. This server deals with any queries from our locally administered handle space, by returning the data associated with a handle to the user who made the query. Our server is also part of the global Handle network, so if it receives a request to resolve a handle administered by some other naming authority, it can potentially hand off the request to an appropriate server somewhere else in the world.

At Nottingham we are creating handles for each document in our test corpus – the EP-odd archive (see section 3). We are also, as part of our test domain, creating handles for all documents referenced from within papers that are part of the corpus – papers from other publications, for example, that are referred to in the *References* sections of the papers we host. In most cases these handles will resolve to a metadata descriptor for the referenced item, but in the case of intra-corpus references i.e. a reference to another paper in the EP-odd archive, it is possible to resolve the handle further and to deliver the document itself.

In a future where most items on the Web possess a URN the preparation of such descriptors would not be necessary – the publisher of the item, be it an individual or a publishing company, would already be maintaining URNs for their resources and ensuring that, at the very minimum, they resolve to correct metadata. Clearly, this is an area that will need careful attention. Standard measures will need to be specified (probably using digital signatures and public key encryption) for validating URNs and their implicit metadata. Only in this way will it be possible to have multiple instances of a URN around the Web and to know that each one is a faithful replica of the original that is guaranteed to be maintained in perpetuity by the owner of the item being referred to. The DOI initiative [6] uses handles for its URNs and in addition to promoting the DOI for e-commerce and rights trading in digital documents is also emphasising to publishers their responsibilities in maintaining the persistence of the DOIs they issue and the integrity of the metadata associated with them.

Dublin Core metadata and RDF

The Dublin Core metadata schema [7], developed from a series of workshops beginning in Dublin, Ohio in 1995, is a set of 15 common metadata elements which, for the most part, can be applied to any object – and not just digital objects. The element set has proved to be useful in many applications – ranging from medicine and libraries (both traditional and digital) to museums. Dublin Core is probably the most widely used metadata schema today.

We chose it for our metadata research because of its simplicity and popularity, but our research is not particular to any one schema – to a large extent, the precise nature of the metadata we are dealing with is irrelevant to the project.

One of the W3C's principal contributions to the metadata field is RDF, which is now a W3C Recommendation for the representation and serialisation of structured metadata in XML format [8]. RDF provides a framework for the representation of any metadata from any schema, to very high levels of complexity and is a cornerstone of the W3C's Semantic Web activities [9].

The RDF model is that of a directed graph with 'objects' as the nodes. Objects in this context can be documents, chapters, organisations, authors or indeed anything that has properties or relations to other objects. The eventual aim is that every concept on the planet can be linked, through relations like these, into one huge, vastly interconnected, graph. It is debatable when, if ever, this will be achieved but RDF is intended to be the framework for this mammoth task.

RDF can, initially, be very daunting and as the complexity of relationships within a given schema increases so also does its readability decline very quickly. However, simple applications, such as the use of RDF to store Dublin Core metadata are far more manageable. We are using RDF for precisely this purpose and thus leaving much of the expressive power of RDF untapped — linking *all* of the resources within a document corpus, with RDF, would be the basis of a much larger project.

3 THE EP-odd ARCHIVE

The Electronic Publishing research group at Nottingham hosts a corpus of documents in PDF format, comprising almost 200 papers from the journal *Electronic Publishing — Origination, Dissemination and Design* (EP-odd). These papers were originally published by John Wiley & Sons Ltd. in the period 1988–1995, and the internally hyperlinked PDF files were the result of the CAJUN project undertaken at Nottingham in 1993 [10]. The PDF files are stored locally, and are also accessible on the Web through the EPRG's website [11].

The EP-odd archive, aside from its value as a public resource, is used within our research group as a test domain for new projects. In 1999, we set out to define descriptive document metadata for every paper in the corpus, allowing the collection to be more easily searched and browsed. Our starting point was a set of metadata for EP-odd that had already been compiled: Nelson Beebe's bibliographies website at the University of Utah [12] contains metadata for a large number of computing journals and publications, including metadata that has been compiled for eight years of EP-odd. This metadata is in BibTeX format and one of our first tasks was to map it to Dublin Core (DC) metadata.

A mapping between the BibTeX and DC schemas was generated by comparing the descriptions of each element in both schemas and matching them as closely as possible. A Perl script performed the conversion of the metadata from one schema to the other, taking the BibTeX as input and producing a set of DC RDF files, one per document, as output. These RDF files were given the same filenames as the PDF files they each represented, but with an additional `.rdf` suffix.

The mapping of BibTeX to DC was not one-to-one. Some of the DC elements had to be filled with generic data (which was usually common across the whole corpus – the Publisher field did not vary, for example), but the mapping was close enough to produce meaningful DC metadata. The end result was a set of simple RDF XML files, each containing descriptive metadata fields for a single PDF file.

These RDF metadata descriptors then formed the basis of a Web search interface to the EP-odd archive. The system was written as a number of Perl/CGI scripts that accessed the RDF files directly. The result is a Web search engine that enables the EP-odd archive to be searched by fields such as title (`DC:Title`) and author (`DC:Creator`), and which displays the metadata (formatted for Web viewing) when the user clicks on a search result in the list. The metadata descriptor pages do, of course, contain a link to the PDF document itself. This system has proved to be a robust and fast interface to the EP-odd archive, and it is publicly accessible at `http://www.cs.nott.ac.uk/~clc/epodd.html`.

Having completed this proof of concept, our focus shifted to the references in the PDFs themselves. Could we hyperlink these so that if users click on a reference in the Acrobat document window, they see metadata for the referenced paper, with the option of getting the whole paper? Many of the references found in the EP-odd archive were to papers from other publications. For these, we often had no electronic copies of the papers, nor did we have any metadata, so we created this metadata manually. Research into metadata inference, for example, pulling out certain fields of metadata from the reference text itself, is continuing, and touches upon important questions such as 'how do we know with confidence that two apparently similar references, in different PDF documents, really do refer to the same target document?'

Our current work is centred on hyperlinking all cited reference within these PDFs by creating handles for every document referred to in the corpus, and assigning metadata to each one. We are developing Acrobat plugins to allow for resolution, in a variety of ways, for the embedded handle links in PDF files. We are also building on the work of the Open Journal Framework project (1995-1998) [13] to achieve the superimposition of hyperlinks into PDF documents where they do not already exist. The situation is complicated by the fact that some of the references already contain absolute links to other papers, which we need either to preserve in some fashion, or to remove entirely.

4 FUTURE PROSPECTS USING XAP

A recent development has opened up exciting new possibilities for metadata within PDF documents. We are referring to XAP, Adobe's metadata initiative, which was unveiled with the release of Acrobat 5.0 in April 2001.

XAP (e**X**tensible **A**uthoring and **P**ublishing) is a framework that allows document authors to add structured metadata directly into a PDF file. XAP was conceived for both the storage of application-specific and general descriptive metadata; its framework is almost a full implementation of the W3C's RDF specification and allows for the embedding of RDF metadata inside PDF documents.

A block of XAP metadata can be assigned to a whole document, and also to individual document components (images, font dictionaries, or any other PDF object). These metadata objects can be stored as uncompressed, unencrypted XML text streams within the file. This, in turn, greatly eases the burden of developing standalone tools to access XAP metadata inside a PDF, because, at least at the level of descriptive metadata for the whole document, it is not necessary to parse the PDF contents in detail – a simple text search for embedded RDF metadata in XML syntax will suffice.

Adding XAP metadata to every document in the EP-odd archive was a natural progression of our continuing work. Since we already have RDF metadata available for every document in the corpus, the task is to insert this data into the PDF files. To this end we are developing a small program which uses the Acrobat System Development Kit (SDK).

The newfound ability to store structured metadata inside a PDF in a standardised way is an exciting one, and will undoubtedly spawn a host of tools and applications that take advantage of it. At least one major search engine already includes normal PDF files in its indexes – it will be interesting to see how soon search engines look for XAP metadata in PDF files given that the process of extracting XAP metadata is much simpler than that of extracting page text from the PDF document.

But the new ability to store this metadata inside a PDF document raises an important issue. Storing document metadata within the document itself (as XAP does) is logical. The metadata *belongs* to the document, and should ideally be as close to it as possible. Users should be able to view the metadata easily, and if they are able to edit the document the metadata should be available in the editing process – in other words the metadata should be *internalised*, and should be a basic part of the document itself.

But we mentioned earlier the need to access metadata without forcing the user into retrieving the whole document file, since the full document may not be what the user wants. Barring complicated server-side solutions where the metadata (and only the metadata) is extracted from the PDF file, and delivered to the user every time it is requested, this means keeping an *external* copy of the metadata. Indeed, when we consider documents that we want to read the metadata for, but which are not available in an electronic form or have distribution restrictions, *externalised* metadata is the only way to make the metadata publicly available.

The issue of metadata duplication is one that raises considerable academic debate, with all sides advancing compelling arguments. How many different copies of metadata should exist for a single object? How are multiple copies to be kept synchronised, and what happens if that synchronisation breaks?

Working on the principle that convenience to the end user is the most important factor, we decided upon an approach of storing the Dublin Core metadata both internally, within the PDFs, *and* externally, as values in the associated handles. This decision raises difficult questions of consistency and synchronisation, and these questions will be a keynote of our future research.

5 CONCLUSIONS

Our system, as developed so far, shows on a small scale the immense possibilities for discovery of resources once one can ‘link everything to everything’, as originally envisaged in the Microcosm project [14] and now re-visited in the idea of the Semantic Web [9]. But, having found a resource, it may not be free to download, and one of the obvious applications for URNs and document metadata is precisely in the area of e-commerce. Indeed, the whole thrust of the DOI initiative is towards e-commerce and Rights management rather than search and discovery [6].

Another avenue for further study is opened up by the possibility of storing component XAP metadata inside a graphically rich format such as PDF. This opens up new possibilities for e-commerce: instead of licensing a whole document for reuse, a customer might be able to purchase the rights to play a movie clip or to reprint just a single image or table.

However, if metadata is to be used to carry out transactions in digital property, this raises the question of trust. How does the user know that the metadata is correct? What is to stop a competitor releasing false metadata records, linked to a URN of some sort, in order to damage an organisation's business or reputation? The need for robust digital signatures, scalable URN implementations and comprehensive 'anti-spoofing' measures are just a few of the many technical issues to be resolved before the next generation of Web technologies can become commonplace.

6 ACKNOWLEDGEMENTS

We are grateful to the National Computing Centre (NCC), Manchester, UK, for funding this research via the award of a PhD studentship to Chris Chapman. Thanks are also due to Adobe Systems Inc. for supplying a pre-release specification of the XAP metadata system.

7 REFERENCES

[1] Tim Bray, "Beyond HTML: XML and Automated Web Processing", *View Source* online magazine (September 1997). http://developer.netscape.com/viewsource/bray_xml.html

[2] Jon Bosak and Tim Bray, "XML and the Second-Generation Web", *Scientific American* **280**, 5, pp. 89–93. (May 1999)

[3] T. Berners-Lee, "Universal Resource Identifiers in WWW: A unifying syntax for the expression of names and addresses of objects on the network as used in the World Wide Web", *World Wide Web Journal* 1, 2 (1996), pp 3–19. See also http://www.faqs.org/rfcs/rfc1630.html (June 1994).

[4] J. Kunze, "The ARK Persistent Identifier Scheme", Internet Draft http://www.ckm.ucsf.edu/people/jak/home/ark-01.txt (March 2001)

[5] Sam X. Sun and Larry Lannom, "Handle System Overview", http://www.handle.net/overview-current.html (August 2000).

[6] "DOI System Overview", http://dx.doi.org/10.1000/203 (2001)

[7] For the latest version of the Dublin Core Schema see http://www.dublincore.org/documents/dces/ (1999)

[8] Ora Lassila, "Introduction to RDF Metadata", http://www.w3.org/TR/NOTE-rdf-simple-intro (1997) See also http://www.w3.org/RDF/FAQ (March 2001) and http://www.w3.org/TR/REC-rdf-syntax (February 1999)

[9] Tim Berners-Lee, James Hendler and Ora Lassila, "The Semantic Web", *Scientific American* **284**, 5 pp. 34–43 (May 2001). See also http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html

[10] Philip N. Smith, David F. Brailsford, David R. Evans, Leon Harrison, Steve G. Probets and Peter E. Sutton, “Journal Publishing with Acrobat: the CAJUN project”, *Electronic Publishing—Origination, Dissemination and Design* **6** (4), 481–493 (December 1993)

[11] See
`http://cajun.cs.nott.ac.uk/compsci/epo/papers/epoddtoc.html`
for the original EP-odd archive and
`http://www.cs.nott.ac.uk/~clc/epodd.html`
for the DC search engine.

[12] Nelson Beebe’s bibliographic archives can be found at:
`http://www.math.utah.edu/~beebe/bibliographies.html`
with a direct link to the EP-odd metadata at:
`http://www.math.utah.edu/pub/tex/bib/epodd.html`

[13] David F. Brailsford, Steve G. Probets, Les Carr and Wendy Hall, ‘Dynamic Link Inclusion in Online PDF Journals’, *Proceedings 7th International Conference on Electronic Publishing (EP98)* pp.550–562. (Springer Verlag 1998)

[14] S. Hitchcock, F. Quek, L. Carr, W. Hall, A. Witbrock and I. Tarr, ‘Linking Everything to Everything: Journal Publishing Myth or Reality?’ *ICCC/IFIP conference on Electronic Publishing 97: New Models and Opportunities*. (1997)